

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Verifying Relevance between Keywords and  
Web site Contents**

**Inventors:**

Benyu Zhang

Hua-Jun Zeng

Zheng Chen

Wei-Ying Ma

Li Li

Ying Li

Tarek Najm

ATTORNEY'S DOCKET NO. MS1-1891US

EV436703188

## **RELATED APPLICATIONS**

[0001] This patent application is related to the following patent applications, each of which are commonly assigned to assignee of this application, and hereby incorporated by reference:

[0002] “Related Term Suggestion for Multi-Sense Query”, filed on 04/15/04;

[0003] U.S. Patent Application no. *<to be assigned>*, titled “Reinforced Clustering of Multi-Type Data Objects for Search Term Suggestion”, filed on 04/15/04; and

[0004] U.S. Patent Application no. 10/427,548, titled “Object Clustering Using Inter-Layer Links”, filed on 05/01/2003.

## **TECHNICAL FIELD**

[0005] Systems and methods of the invention pertain to data mining.

## **BACKGROUND**

[0006] A keyword or keyphrase is a word or set of terms submitted by a Web surfer to a search engine when searching for a related Web page/site on the World Wide Web (WWW). Search engines determine the relevancy of a Web site based on the keywords and keyword phrases that appear on the page/site. Since a significant percentage of Web site traffic results from use of search engines, Web site promoters know that proper keyword(s) selection is vital to increasing site traffic to obtain desired site exposure. Techniques to identify keywords relevant to a Web site for search engine result optimization include, for example,

evaluation by a human being of Web site content and purpose to identify relevant keyword(s). This evaluation may include the use of a keyword popularity tool. Such tools determine how many people submitted a particular keyword or phrase including the keyword to a search engine. Keywords relevant to the Web site and determined to be used more often in generating search queries are generally selected for search engine result optimization with respect to the Web site.

[0007] After identifying a set of keywords for search engine result optimization of the Web site, a promoter may desire to advance a Web site to a higher position in the search engine's results (as compared to displayed positions of other Web site search engine results). To this end, the promoter bids on the keyword(s) to use with specific URL(s), wherein the bidding indicates how much the promoter will pay each time a Web surfer clicks on the promoter's listings associated with the keyword(s). In other words, keyword bids are pay-per-click bids for specific URL (Web site) promotion. The larger the amount of the keyword bid as compared to other bids for the same keyword, the higher (more prominently with respect to significance) the search engine will display the associated Web site in search results based on the keyword. Unfortunately, advertiser bidding terms may not be relevant to the Web site contents and, as a result, may not match the terms or language used by an end-user.

[0008] It may appear that the simplest way to verify a keyword(s) against a Web site (i.e., Web site content) is to use a conventional retrieval approach, which measures the similarity only between the keyword(s) and the Web site, without any additional data point comparisons. However, this technique is substantially limited. Even though the keyword(s) may be related to the Web site, the Web site

itself may not include threshold criteria (e.g., direct match, number of occurrences, etc.) supporting the desired keyword(s), causing rejection of a potentially valuable bidding term. For example, consider that an online shopping corporation with an associated Web site bids on the phrase “online shopping”. If the conventional retrieval approach is used and a relatively small number of occurrences of the keyword “shopping” and no occurrence of keyword “online” are found in the Web site, the potentially valuable keyphrase of “online shopping” may be mistakenly disqualified as a bidding term.

[0009] Another conventional technique is to classify a submitted bid term/phrase and Web site to obtain two categories probabilities vectors, which are then combined into a final relevance score. The problem with this conventional technique is that it does not evaluate the term/phrase directly against his website, which can be substantially problematic. For example, if an advertiser bids on the term “Italian shoes”, and his website sells shoes but NOT Italian shoes, the conventional classification technique will indicate to the advertiser that the bid phrase of “Italian shoes” is irrelevant to the Web site.

[0010] In view of the above, systems and methods to better identify keywords relevant to Web site content would be welcomed by Web site promoters. This would allow the promoters to bid terms more likely to be used by an end-user. Ideally, these systems and methods would be independent of the need for a human being to evaluate Web site content to identify relevant keywords for search engine optimization and keyword bidding.

## **SUMMARY**

[0011] Systems and methods for verifying relevance between terms and Web site contents are described. In one aspect, site contents from a bid URL are retrieved. Expanded term(s) semantically and/or contextually related to bid term(s) are calculated. Content similarity and expanded similarity measurements are calculated from respective combinations of the bid term(s), the site contents, and the expanded terms. Category similarity measurements between the expanded terms and the site contents are determined in view of a trained similarity classifier. The trained similarity classifier having been trained from mined web site content associated with directory data. A confidence value providing an objective measure of relevance between the bid term(s) and the site contents is determined from the content, expanded, and category similarity measurements evaluating the multiple similarity scores in view of a trained relevance classifier model.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

[0012] In the figures, the left-most digit of a component reference number identifies the particular figure in which the component first appears.

[0013] Fig. 1 illustrates an exemplary system for verifying relevance between terms and Web site contents.

[0014] Fig. 2 illustrates an exemplary procedure for verifying relevance between terms and Web site contents.

[0015] Fig. 3 illustrates an exemplary procedure for verifying relevance between terms and Web site contents. In particular, Fig. 3 is a continuation of the exemplary operations of Fig. 2.

[0016] Fig. 4 shows an exemplary suitable computing environment on which the subsequently described systems, apparatuses and methods for verifying relevance between terms and Web site contents may be fully or partially implemented.

## **DETAILED DESCRIPTION**

### **Overview**

[0017] The following systems and methods verify relevance between terms and Web site contents to address the limitations of conventional term qualification techniques. To this end, the systems and methods combine multiple similarity measurements via trained classifier models to provide a single confidence value indicating whether bid term(s) is/are relevant to a particular Web site's content. More particularly, and in this implementation, the multiple similarity measurements include content, category, and proper name similarity scores.

[0018] Content similarity scores include direct and expanded content similarity. Direct content similarity is determined by evaluating vector models of the bid term(s) and site contents of the submitted Web site. Expanded similarity is determined by evaluating similarity between vector models of expanded terms and the site contents. The expanded terms are mined from a search engine in view of high-frequency of occurrence historical query terms, and determined to be semantically and/or contextually similar to the bid term(s). Category similarity is determined by applying a trained similarity categorization (classifier) model to the expanded terms and Web site contents to determine relatedness of categories between these inputs. Proper name similarity is determined by evaluating the bid term(s) and Web site contents in view of a database of proper names. These

multiple similarity measurements are combined using a combined relevance classifier model that is trained to generate a single relevance confidence value from these score in view of an accept/reject threshold. The confidence value provides an objective measurement of the relevance of the bid term(s) to the Web site in view of these multiple different similarity measurements.

[0019] These and other aspects of the systems and methods for verifying relevance between terms and Web site contents are now described in greater detail.

### **An Exemplary System for Editorial Verification**

[0020] Turning to the drawings, wherein like reference numerals refer to like elements, the systems and methods for verifying relevance between terms and Web site contents are described and shown as being implemented in a suitable editorial verification computing environment. Although not required, the invention is described in the general context of computer-executable instructions (program modules) being executed by a personal computer. Program modules generally include routines, programs, objects, components, data structures, etc., that perform particular tasks or implement particular abstract data types. While the systems and methods are described in the foregoing context, acts and operations described hereinafter may also be implemented in hardware.

[0021] Fig. 1 shows system 100 for verifying relevance between bid terms and bid Web site contents. In this implementation, system 100 includes editorial verification server 102 coupled across a network 104 to search engine 106. Network 104 may include any combination of a local area network (LAN) and

general wide area network (WAN) communication environments, such as those which are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet. Editorial verification server 102 includes a number of program modules 108, for example, search term suggestion (STS) module 110, relevance verification module 112, classification module 114, term matching module 116, and other program module(s) 118 such as a Web page crawler to retrieve site contents from a bid Universal Resource Locator (URL) identifying a Web site.

[0022] An end-user (e.g., an advertiser, Web site promoter, etc.) provides bid input 120 to editorial verification server 102 for relevance verification of bid term(s) to site content of a bid URL. Bid input 120 includes bid term(s) 122 and bid URL 124. In one implementation, editorial verification server 102 includes one or more user input interfaces (e.g., see user input interfaces 460 of Fig. 4) such as a keyboard, mouse, voice recognition system, and/or the like, for the end-user to supply bid input 120 to editorial verification server 102. In another implementation, editorial verification server 102 is coupled across network 104 to a client computing device (e.g., remote computer 480 of Fig. 4) for the end-user to provide bid input 120 to editorial verification server 102.

### Exemplary Search Term Suggestion

[0023] Responsive to receiving bid input 120 from an end-user, search term suggestion module 102 generates search term suggestion list 126 to expand term(s) 122 with semantically and/or contextually related terms. Multiple senses or contexts of a term 122 may provide additional term meaning, as described below. TABLE 1 shows an exemplary suggested term list 126 of terms determined to be related to a term(s) 122 of “mail.” Terms related to term(s) 122 are shown in this example in column 1, titled “Suggested Term(s)”.

**TABLE 1**  
**AN EXEMPLARY SUGGESTED TERM LIST FOR**  
**THE BID TERM “MAIL”**

<b>Suggested Term(s)</b>	<b>Similarity</b>	<b>Frequency</b>	<b>&lt;Context &gt;</b>
hotmail	0.246942	93161	online e-mail related
yahoo	0.0719463	165722	
mail.com	0.352664	1455	
yahoo mail	0.0720606	39376	
www.mail.com	0.35367	711	
email.com	0.484197	225	
www.hot	0.186565	1579	
www.msn.com	0.189117	1069	
mail.yahoo.com	0.0968248	4481	
free email	0.132611	1189	
www.aolmail.com	0.150844	654	
check mail	0.221989	66	
check email	0.184565	59	
msn passport	0.12222	55	
www.webmail.aol.com	0.0800538	108	
webmail.yahoo.com	0.08789	71	
free email account	0.0836481	65	

<b>Suggested Term(s)</b>	<b>Similarity</b>	<b>Frequency</b>	
mail	1	2191	Traditional mail related
usps	0.205141	4316	
usps.com	0.173754	779	
united parcel service	0.120837	941	
postal rates	0.250423	76	
stamps	0.156702	202	
stamp collecting	0.143618	152	
state abbreviations	0.104614	300	
postal	0.185255	66	
postage	0.180112	55	
postage rates	0.172722	51	
usps zip codes	0.136821	78	
us postmaster	0.109844	58	

[0024] Referring to TABLE 1, note that for each suggested term (col. 1), search term suggestion list 126 also includes a respective similarity measurement value (see, col. 2) to indicate relatedness between the suggested term(s) and term(s) 122, and a respective frequency of use score (see, col. 3) to provide an indication of how frequently the suggested term(s) of col. 1 has/have been submitted to the search engine 106. In this example, each similarity value of col. 2 provides a similarity measure or score between a corresponding suggested term (col. 1) and bid term(s) 122, which is “mail” in this example. Each frequency value, or score, indicates the number of times that the suggested term was used by a particular search engine 106 in an end-user search query. If to be presented to an end-user, the suggested term list 126 is sorted as a function of business goals, for instance by suggested term(s), similarity, and/or frequency scores.

[0025] Any given term(s) 122 (e.g., mail, etc.) may have more than a single context within which the bid term may be used. To account for this, search term suggestion module 110 segregates suggested term(s) by context. For example,

referring to TABLE 1, the term(s) 122 of “mail” has two (2) contexts: (1) traditional off-line mail and (2) online e-mail. Note that a respective (segregated or independent) list of suggested term(s) is shown for each of these two bid term contexts.

[0026] Suggested term(s) of suggested term list 126 may be more than synonyms of term(s) 122. For instance, referring to TABLE 1, a suggested term of “usps” is an acronym for an organization that handles mail, not a synonym for a bid term of “mail.” However, “usps” is also a term very contextually related to a “mail” bid term, and thus, is shown in the suggested term list 126. In one implementation, search term suggestion module 110 determines the relationship between a related term *R* (e.g. “usps”) and a target term *T* (e.g. “mail”) as a function of the following association rule:  $\text{itr}(T) \rightarrow \text{itr}(R)$ , wherein “itr” represents “interested in”. That is, if an end-user (advertiser, Web site promoter, and/or the like) is interested in *R*, then the end-user will likely also be interested in *T*.

[0027] To generate search term suggestion list 126, search term suggestion module 110 submits select ones of historical queries mined from query log 130 to search engine 106. The select ones of the historical queries for submission to search engine 105 identified by search term suggestion module 126 as having a substantially high frequency of occurrence (FOO) as compared to other ones of the historical query terms mined from query log 130. In this implementation, a configurable threshold value is used to determine whether a historical query has a relatively higher or low frequency of occurrence. For example, historical query terms that occur at least a threshold number of times are said to have a high frequency of occurrence. Analogously, historical query terms that occur less than

the threshold number of time are said to have a low frequency of occurrence. For purposes of illustration, such a threshold value is shown as a respective portion of “other data” 132. High and low FOO query terms are shown as “high/low FOO queries” portion of “other data” 132.

[0028] Search term suggestion module 110 extracts a set of features or snippet descriptions from select ones of the returned search results (e.g., one or more top-ranked search results) for each query term. Search term suggestion module 110 performs text preprocessing operations on the extracted data to generate individual term tokens. To reduce dimensionality of the tokens, search term suggestion module 110 removes any stop-words (e.g., “the”, “a”, “is”, etc.) and removes common suffixes, and thereby normalizes the terms, for example, using a known Porter stemming algorithm. Search term suggestion module 110 arranges the resulting terms and other extracted features into one or more search term suggestion (STS) vectors (shown as a respective portion of term vectors 134). Each STS vector 134 has dimensions based on term frequency and inverted document frequency (TFIDF) scores.

[0029] A weight for the  $i^{\text{th}}$  vector's  $j^{\text{th}}$  term is calculated as follows:

$$w_{ij} = TF_{ij} \times \log(N / DF_j)$$

wherein  $TF_{ij}$  represents term frequency (the number of occurrences of term  $j$  in the  $i^{\text{th}}$  record),  $N$  is the total number of query terms, and  $DF_j$  is the number of records that contain term  $j$ . Search term suggestion module 110 uses these respective weights to group similar terms and context from STS vectors 134 to generate term clusters 136. To this end, and in this implementation, given the vector

representation of each term, a cosine function is used to measure the similarity between a pair of terms (recall that the terms were normalized):

$$sim(q_j, q_k) = \sum_{i=1}^d w_{ij} \cdot w_{ik}$$

Thus, the distance between the two terms (a similarity measurement) is defined as:

$$dist(q_j, q_k) = 1 - sim(q_j, q_k)$$

Such search term suggestion (STS) similarity measurements are shown as a respective portion of “other data” 132. Exemplary such similarity values are shown above in an exemplary suggested term list 126 of TABLE 1.

[0030] Search term suggestion module 110 uses the calculated term similarity measurement(s) to cluster/group terms in the STS vectors 134 into a high FOO historical query term based portion of term cluster(s) 136. More particularly, and in this implementation, search term suggestion module 110 uses a known density-based clustering algorithm (DBSCAN) to generate these term cluster(s) 136. DBSCAN uses two parameters: *Eps* and *MinPts*. *Eps* represents a maximum distance between points in a term cluster 136. A point is a feature vector of a term. In a high dimensional space, vectors are equivalent to points. *MinPts* represents a minimum number of points in a term cluster 136. To generate a cluster 136, DBSCAN starts with an arbitrary point *p* and retrieves all points density-reachable from *p* with respect to *Eps* and *MinPts*. If *p* is a core point, this operation yields a term cluster 136 with respect to *Eps* and *MinPts*. If *p* is a border point, no points are density-reachable from *p* and DBSCAN visits the next point.

[0031] Search term suggestion module 110 then compares term(s) 122 to respective ones of the term(s) in the term clusters 136. Since the term clusters

include features that are semantically and/or contextually related to one another, term(s) 122 is/are evaluated in view of multiple related contexts, or “senses” to expand term(s) 122, and thereby provide generate search term suggestion list 126. In one implementation, if search term suggestion module 110 determines that term(s) 122 match term(s) from no more than a single cluster 136, search term suggestion module 110 generates suggested term list 126 from the single cluster 136. In this implementation, a match may be an exact match or a match with a small number of variations such as singular/plural forms, misspellings, punctuation marks, etc. The generated term list is ordered by a certain criteria, which, for example, could be a linear combination of the FOO and the similarity between term(s) 122 and the suggested terms, as:

$$\text{Score}(q_i) = \alpha \text{FOO}(q_i) + \beta \text{sim}(q_i, Q)$$

where  $\alpha + \beta = 1$ .

[0032] If search term suggestion module 110 determines that term(s) 122 match term(s) in multiple term clusters 136, search term suggestion module 110 generates suggested term list 126 from terms of the multiple term clusters. The suggested terms from each clusters are ordered using the same method as described above in paragraph no. [0031].

[0033] An exemplary system and method for search term suggestion module 110 to generate search term suggestion list 126 is described in U.S. Patent Application serial no. *<to be supplied>*, titled “Related Term Suggestion for Multi-Sense Query”, filed on 04/15/04.

#### Exemplary Relevance Verification

[0034] Relevance verification module 112 uses the suggested term(s) (terms that expand bid input 120 terms(s) 122) of search term suggestion list 126 and bid

input 120 (i.e., term(s) 122 and site content from URL 124) to generate confidence value 138, which measures relevance between the bid term(s) 122 and site contents of the bid URL 124. To this end, relevance verification module 112 calculates confidence value 138 from multiple similarity measurements, which for purposes of illustration and discussion are shown as relevance verification (RV) similarity measurements 140. In this implementation, RV-similarity measurements 140 include, for example, content similarity, classification similarity, and proper name similarity scores. Each of these types of RV-similarity measurements 140 is now described.

[0035] Content similarity measurements portion of RV-similarity measurements 140 include direct and expanded similarity measurements. To calculate direct similarity, relevance verification module 112 measures similarity/relatedness between term(s) 122 and site contents of URL(s) 13, both being modeled in vector space. To calculate expanded similarity, site contents of URL 124 are retrieved, for example, by a Web page crawler module, which is represented by a respective portion of “other program module(s)” 118. Relevance verification module 112 determines similarity between suggested term(s) of search term suggestion list 126 and site contents of URL 124, both inputs also having been modeled in vector space. As described above, the suggested term(s) of the search term suggestion list 126 were: (a) mined from results returned by search engine 106 in view of submitted high FOO historical query terms. Thus, the suggested term(s) is/are determined to be semantically and/or contextually related to the bid term(s) 122.

[0036] The proper name similarity measurements portion of RV-similarity measurements 140 indicates similarity/relatedness between any proper name(s) detected in the bid term(s) 122 and site contents of URL 124. For purpose of discussion, a database of proper names is represented with a respective portion of “other data” 132. Such proper names include, for example, names of countries, cities and famous trademarked brands. More particularly, upon detecting any proper names in the bid input 120, relevance verification module 112 calculates proper name similarity as:

$\text{Prop\_Sim}(\text{term}, \text{page}) =$

- 1 - If a *term* contains a proper name *P*, and *page* contained a conformable proper name *Q*.
- 0 - If *term* contains a proper name *P*, and *page* only contains unconformable proper name(s) *Q*.
- 0.5 – Otherwise.

A proper name is conformable with itself and its ancestors. For example, a low-level geographical location is conformable with a high-level geographical location which contains it, e.g. Milan is conformable with Italy.

[0037] The classification similarity measurements portion of RV-similarity measurements 140 measure relatedness between suggested term(s) of the search term suggestion list 126 and site contents of URL 124. More particularly, classification similarity measurements are generated by submitting the suggested terms and Web site contents to trained similarity classifier (categorization) 142. Relevance verification module 122 trains similarity classifier 142 with any one of

a number of different classification techniques (e.g., naïve Bayesian (NB), support vector machine (SVM), statistical n-gram based naïve Bayesian (N-Gram), nearest neighbor (KNN), decision tree, co-training, boosting, and/or the like), as is now described.

#### Exemplary Offline Similarity Classifier Training

[0038] Relevance verification module 112 trains similarity classifier 142 as  $\Phi: X \mapsto L$  on directory data (see, “other data” 132), wherein  $X$  is input (a string stream with scale from a single term to several web page contents), and  $L$  is output (a probability over all the top2 levels of categories). The category taxonomy is of hierarchical structure. In this implementation, we use the 2<sup>rd</sup>-level categories of LookSmart® directory data, the sum of these categories is some number (e.g., 74), for the classification. Relevance verification module 112 performs feature extraction and feature selection operations on the directory data. More particularly, relevance verification module 112 extracts snippet descriptions (extracted data) from Web page(s) identified by the directory data. The Web page(s) are retrieved, for example, by a Web page crawler module represented with a respective portion of “other program module(s)” 118. Each snippet description for a particular Web page includes, for example, one or more of a title, metadata, body, anchor text, font size, hyperlinks, images, raw HTML (e.g., summarization and page layout information), and/or the like.

[0039] Relevance verification module 112 applies simple text preprocessing to generate linguistic tokens (i.e., tokenizes individual terms) from the extracted features/data. To reduce dimensionality of the tokens, relevance verification module 112 removes any stop-words and removes common suffixes to normalize

the terms, for example, using a known Porter stemming algorithm. Relevance verification module 112 arranges the resulting extracted features into one or more relevance verification (RV) term vectors (i.e., RV-vectors 134). As such, each Web page is represented as a feature vector, whose element is a word with its weighting  $x_i = \langle x_{i1}, x_{i2} \dots x_{in} \rangle$ . The weighting  $x_{ij}$  is calculated by length-normalized  $\log(\text{tf}).\text{idf}$ , which has the form:

$$\text{idf}_t \times \frac{1 + \log(f_{d,t})}{1 + \log(\text{avef}_d)} \times \frac{1}{\text{avedlb} + S \times (dlb_d - \text{avedlb})},$$

where,  $d$  represents the original document,  $t$  represents term,  $f_{d,t}$  represents frequency of term  $t$  in  $x$ ,  $\text{idf}_t$  represents inverse document frequency of term  $t$ ,  $dlb_x$  represents number of unique terms in  $x$ ,  $\text{avef}_x$  represents average of term frequencies in  $x$ , and  $\text{avedlb}$  represents average of  $dlb_x$  in the collection.

[0040] Feature selection operations of relevance verification module 112 further reduce the features of RV-vectors 134 (too many features can degrade the performance and accuracy of a classification system). In this implementation, information gain (IG) selection method is used for feature selection. Information gain of a term measures the number of bits of information obtained for category prediction by the presence or absence of the term in a document as follows:

$$IG(t) = -\sum_{i=1}^m p(c_i) \log p(c_i) + p(t) \sum_{i=1}^m p(c_i | t) \log p(c_i | t) + p(\bar{t}) \sum_{i=1}^m p(c_i | \bar{t}) \log p(c_i | \bar{t}),$$

wherein,  $t$  represents term,  $c$  represents category, and  $m$  represents total number of categories. Other feature selection methods, such as mutual information (MI), document frequency (DF), and Linear Discriminant Analysis (LDA), can also be used.

[0041] In this implementation, relevance verification module 112 classifier training operations employ statistical n-gram model based Naïve Bayesian classifier (n-gram), although other types of classifiers could be used. In particular, different from Naïve Bayesian classifier, statistical n-gram model doesn't assume the independent of word stream. It assumes Markov n-gram independence, i.e. one word is dependent to previous n-1 words according to:

$$p(w_i | w_1, w_2, \dots, w_{i-1}) = p(w_i | w_{i-n+1}, \dots, w_{i-1}).$$

A straightforward estimation of this probability from a training corpus is given by the observed frequency of:

$$p(w_i | w_{i-n+1}, \dots, w_{i-1}) = \frac{\#(w_{i-n+1}, \dots, w_i)}{\#(w_{i-n+1}, \dots, w_{i-1})}.$$

[0042] Most of the  $\#(w_{i-n+1}, \dots, w_i)$ ,  $\#(w_{i-n+1}, \dots, w_{i-1})$  values are zero in the training data. So smoothing technology is proposed to estimate the zero probability to deal with any data sparseness. Back-off n-gram model is one way to deal with this issue, as follows:

$$p(w_i | w_{i-n+1}, \dots, w_{i-1}) = \begin{cases} \hat{p}(w_i | w_{i-n+1}, \dots, w_{i-1}), & \text{if } \#(w_{i-n+1}, \dots, w_i) > 0 \\ \beta(w_{i-n+1}, \dots, w_{i-1}) \times p(w_i | w_{i-n+2}, \dots, w_{i-1}), & \text{otherwise} \end{cases}$$

wherein,

$$\hat{p}(w_i | w_{i-n+1}, \dots, w_{i-1}) = \frac{\text{discount} \#(w_{i-n+1}, \dots, w_i)}{\#(w_{i-n+1}, \dots, w_{i-1})}$$

is the discounted conditional probability, and  $\beta(w_{i-n+1}, \dots, w_{i-1})$  is the back-off factor to back-off n-gram to (n-1)-gram:

$$\beta(w_{i-n+1}, \dots, w_{i-1}) = \frac{1 - \sum_{x: \#(w_{i-n+1}, \dots, w_{i-1}, x) > 0} \hat{p}(x | w_{i-n+1}, \dots, w_{i-1})}{1 - \sum_{x: \#(w_{i-n+1}, \dots, w_{i-1}, x) > 0} \hat{p}(x | w_{i-n+2}, \dots, w_{i-1})}.$$

[0043] There are several algorithms to calculate the discounted probability.

In this implementation, “absolute smoothing” is used as follows:

$$\hat{p}(w_i | w_{i-n+1}, \dots, w_{i-1}) = \frac{\#(w_{i-n+1}, \dots, w_i) - b}{\#(w_{i-n+1}, \dots, w_{i-1})}$$

where

$$b = \frac{n_1}{n_1 + 2n_2}, \text{ and } n_i$$

is the number of words which occur exactly  $i$  times in training data. Thus, we can modify the NB classifier as n-gram classifier:

$$c_{n\text{-gram}} = \arg \max_{c_j \in V} p(c_j) \prod_i p_{c_j}(w_i | w_{i-n+1}, \dots, w_{i-1})$$

In this implementation,  $n=3$ , and the n-gram classifier is called the 3-gram classifier.

### Expert Combination of Similarity Measurements

[0044] Relevance verification module 112 evaluates the multiple RV-similarity measurements 140 in view of combined relevance classifier 144 to generate confidence value 138, which indicates an objective relevance of the bid term(s) 122 to site content of bid URL 124. Combined relevance classifier 144 is trained with supervised learning, for example, as an SVM classifier, with data in the form of <term(s), Web page (URL), Accept/Reject> in view of a reject/accept term/phrase threshold. For purposes of discussion, a reject/accept term threshold is shown as a respective portion of “other data” 132.

[0045] RV-similarity measurements 140 are treated as a feature vector for bid input 120 (i.e., a <term, page> pair). For purposes of illustration and discussion, RV-similarity measurements (SM) as feature vector(s) are shown as RVSM feature vector(s) 140. We have the following bid input 120 and RV-similarity measurement 140 calculations:

- bid input 120: <term(s) 122, URL 124>;
- content based RV-similarity measurements 140 of term(s) 122, URL 124, which is represented as Sim(term(s) 122, URL 124);
- expanded content based RV-similarity measurements 140 – Ex\_Sim(expanded term(s) 126, URL 124);
- similarity classifier 142 based RV-similarity measurements 140 – Cate\_Sim(category of expanded terms 126, category of URL); and
- proper name based RV similarity measurements 140 – Proper\_Sim(proper names, term(s) 122, URL 124).

[0046] Relevance verification module 112 applies RVSM feature vector(s) 140 of <term, query> to combined relevance classifier 144 to map the multiple RV-similarity values 140 in view of the reject/accept relevance threshold to calculate respective RV-similarity type weights (i.e., content, expanded, category, and proper similarity measurement types) and the final confidence value 138.

#### Classification of Low FOO Terms

[0047] In view of a configurable threshold, if confidence value 138 indicates that term(s) 122 should be rejected as irrelevant to site contents of URL 124, classification module 114 generates suggested term list 126 based on low frequency of occurrence (FOO) query terms for the end-user to evaluate in view of the site contents of URL 124. In this implementation, the suggested term list 126 is shown as message 146 being communicated to an end-user for evaluation. In particular, classification module 114 uses STS classifier 148 from term clusters 136, which as described above, were generated from high frequency of occurrence (FOO) query log terms. Classification module 114 uses STS classifier 148 to group the high FOO-based term clusters 136 into one or more STS categories (see, "other data" 132) as a function on their respective term content. Term clusters 136 are already in a vector space model suitable for classification operations. Additionally, stop-word removal and word stemming (suffix removal) has already reduced dimensionality of term cluster 136 content. In one implementation, additional dimensionality reduction techniques, for example, feature selection or re-parameterization, may be employed.

[0048] In this implementation, to classify a class-unknown term cluster 136, Classification module 114 uses the  $k$ -Nearest Neighbor classifier algorithm to rank the class-unknown cluster's neighbors among the term vectors, and uses the class labels of the  $k$  most similar neighbors to predict the class of the class-unknown term. The classes of these neighbors are weighted using the similarity of each neighbor to  $X$ , where similarity is measured by Euclidean distance or the cosine value between two document vectors. The cosine similarity is as follows:

$$sim(X, D_j) = \frac{\sum_{t_i \in (X \cap D_j)} x_i x_{dij}}{\|X\|_2 \|D_j\|_2}$$

where  $X$  is the test document, represented as a vector;  $D_j$  is the  $j$ th training document;  $t_i$  is a word shared by  $X$  and  $D_j$ ;  $x_i$  is the weight of term  $t_i$  in  $X$ ;  $d_{ij}$  is the weight of term  $t_i$  in document  $D_j$ ;  $\|X\|_2 = \sqrt{x_1^2 + x_2^2 + x_3^2 \dots}$  is the norm of  $X$ , and  $\|D_j\|_2$  is the norm of  $D_j$ . A cutoff threshold is used to assign the new document to a known class.

[0049] In another implementation, a different statistical classification and machine learning technique (e.g., including regression models, Bayesian classifiers, decision trees, neural networks, and support vector machines) other than a nearest-neighbor classification technique is used to generate the trained STS classifier.

[0050] Classification module 114 submits low frequency of occurrence (FOO) query terms (see, high/low query terms portion of "other data" 132), one-by-one, to search engine 106. Responsive to receiving corresponding search results for each search engine submitted query, and using techniques already

described, Classification module 114 extracts features such as a snippet description from each of the one or more retrieved Web page(s) identified in the search results. In this implementation, features are extracted from a first top-ranked Web page. These extracted features are represented in a respective portion of “other data” 132. In another implementation, features are extracted from a multiple top-ranked Web pages. For each retrieved and parsed Web page, Classification module 114 stores the following information in a respective record of extracted features: the snippet description, the search query used to obtain the retrieved Web page, and a Universal Resource Identifier (URI) of the retrieved Web page. Next, Classification module 114 tokenizes, reduces dimensionality, and normalizes extracted features 138 derived from low FOO query terms to generate another set of term vectors (i.e., STS-vectors 134).

[0051] Classification 114 clusters the term(s) in STS-vectors 134 into a respective set of term clusters 136, which are clusters based on low FOO query terms. This clustering operation is performed using the trained STS classifier 148, which as described above, was generated from high FOO query terms. Classification module 114 evaluates term(s) in view of these term clusters to identify and return a suggested term list 126 comprising these other terms to the end-user.

### Exemplary Term Matching

[0052] In view of a configurable threshold, if confidence value 138 indicates that term(s) 122 should be accepted as irrelevant to site contents of URL 124, the bid input 120 is stored in bidding database 150 for resolution of subsequent queries 152 received from an end-user. For instance, responsive to

receiving query 152 from an end-user searching for a Web page, term match module 116 edits the distance between the term(s) in query 150 and the term(s) from bidding database 150 to determine relevance of term(s) in query 152 to bid term(s) 122. In particular, term match module 116 determines relevance as follows:

$$relevance^* = \frac{\log(1 + \alpha(\beta \times fCommon + (1 - \beta) \times fDistance))}{\log(1 + \alpha)},$$

wherein fCommon represents the number of common term(s), and fDistance represents the number of times the bid term(s) 122 have been exchanged with term(s) of query 152.

### **An Exemplary Procedure**

[0053] Fig. 2 illustrates an exemplary procedure 200 for verifying relevance between terms and Web site contents. For purposes of discussion, operations of the procedure are discussed in relation to the components of Fig. 1. (All reference numbers begin with the number of the drawing in which the component is first introduced). At block 202, search term suggestion module 110 generates a first set of term clusters 136 from search engine 106 search results. For purposes of discussion, such search results are shown as a respective portion of "other data" 132. To obtain the search results, search term suggestion module 110 communicates high frequency of occurrence historical queries mined from query log 130. The term clusters 136 include snippet descriptions, corresponding search queries, and Web pages determined by the search term suggestion module 110 to be semantically and/or contextually related to the submitted high frequency of occurrence historical queries.

[0054] At block 204, responsive to editorial verification server 102 receiving bid input 120, including term(s) 122 and URL 124, search term suggestion module 110 identifies expanded terms from the term clusters 136 generated from the high frequency of occurrence historical query terms. Such expanded terms included terms that are semantically and/or contextually related to term(s) 122 and/or site content of the bid URL 124. Expanded terms are shown as suggested term list 126 of Fig. 1. At block 206, relevance verification module 112 calculates content, expanded, classified, and proper name similarity values (i.e., RV-similarity measurements 140) respectively from combinations of bid term(s) 122, bid URL 124, expanded terms of suggested term list 126, a trained similarity classifier 142, and/or a database of proper names. At block 208, relevance verification module 112 combines RV-similarity measurements 140 in view of a trained combined relevance classifier 144 and an accept/rejected threshold value (see, "other data" 132) to obtain a confidence value 138. Confidence value 138 provides an objective measurement of the relevance between the bid term(s) 122 and the bid URL 124.

[0055] At block 210, relevance verification module 112 determines whether the confidence value 138 is too low in view of the accept/rejected threshold. If so, the procedure continues at block 212. At block 212, classification module 114 generates suggested term list 126 from a second set of term clusters 136 based on search engine 106 results to low FOO historical queries and a classifier trained on the first set of term clusters 136. Term(s) of the suggested term list 126 are determined by classification module 114 to be semantically and/or contextually similar to site contents associated with the bid URL 124. For purposes of

illustration, the classifier is shown as STS classifier 148. In this example, suggested term list 126 is shown as being communicated as a message 146 to an end-user for evaluation.

[0056] At block 208, if relevance verification module 112 determines that the confidence value 138 is acceptable (not too low in view of the accept/rejected threshold), the procedure continues at block 302 of Fig. 3, as indicated by on-page reference “A”.

[0057] Fig. 3 illustrates an exemplary procedure 300 for verifying relevance between terms and Web site contents. In particular, Fig. 3 is a continuation of the exemplary operations of Fig. 2. At block 302, relevance verification module 112 stores/caches the bid term(s) 122 and bid URL 124 into bidding database 150. At block 304, Responsive to receipt by editorial verification server 102 of any end-user query 152, term matching module 116 determines if terms of the search query 152 are relevant to the term(s) 122 stored in the bidding database 150 in view of a possibility that the query terms may not exactly match the bid term(s) 122. At block 306, if the term(s) of query 152 are determined to be relevant to bid term(s) 122, editorial verification server 102 communicates the corresponding bid URL 124 to the end-user as a search result.

### **An Exemplary Operating Environment**

[0058] Fig. 4 illustrates an example of a suitable computing environment 400 on which the system 100 of Fig. 1 and the methodology of Figs. 2 and 3 for verifying relevance between terms and Web site contents may be fully or partially implemented. Exemplary computing environment 400 is only one example of a suitable computing environment and is not intended to suggest

any limitation as to the scope of use or functionality of systems and methods the described herein. Neither should computing environment 400 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in computing environment 400.

[0059] The methods and systems described herein are operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well-known computing systems, environments, and/or configurations that may be suitable for use include, but are not limited to, personal computers, server computers, multiprocessor systems, microprocessor-based systems, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and so on. Compact or subset versions of the framework may also be implemented in clients of limited resources, such as handheld computers, or other computing devices. The invention is practiced in a distributed computing environment where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

[0060] With reference to Fig. 4, an exemplary system for verifying relevance between terms and Web site contents includes a general purpose computing device in the form of a computer 410. The following described aspects of computer 410 are exemplary implementations of client computing device PSS server 102 (Fig. 1) and/or client computing device 106. Components of computer 410 may include, but are not limited to, processing unit(s) 420, a system

memory 430, and a system bus 421 that couples various system components including the system memory to the processing unit 420. The system bus 421 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example and not limitation, such architectures may include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

[0061] A computer 410 typically includes a variety of computer-readable media. Computer-readable media can be any available media that can be accessed by computer 410 and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer-readable media may comprise computer storage media and communication media. Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer 410.

[0062] Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism, and includes any information delivery media. The term “modulated data signal” means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example and not limitation, communication media includes wired media such as a wired network or a direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of the any of the above should also be included within the scope of computer-readable media.

[0063] System memory 430 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 431 and random access memory (RAM) 432. A basic input/output system 433 (BIOS), containing the basic routines that help to transfer information between elements within computer 410, such as during start-up, is typically stored in ROM 431. RAM 432 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 420. By way of example and not limitation, Fig. 4 illustrates operating system 434, application programs 435, other program modules 436, and program data 437. In one implementation, application programs 435 comprise program modules 108 of Fig. 1. In this same scenario, program data 437 comprises program data 128 of Fig. 1.

[0064] The computer 410 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only,

Fig. 4 illustrates a hard disk drive 441 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive 451 that reads from or writes to a removable, nonvolatile magnetic disk 452, and an optical disk drive 455 that reads from or writes to a removable, nonvolatile optical disk 456 such as a CD ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 441 is typically connected to the system bus 421 through a non-removable memory interface such as interface 440, and magnetic disk drive 451 and optical disk drive 455 are typically connected to the system bus 421 by a removable memory interface, such as interface 450.

[0065] The drives and their associated computer storage media discussed above and illustrated in Fig. 4, provide storage of computer-readable instructions, data structures, program modules and other data for the computer 410. In Fig. 4, for example, hard disk drive 441 is illustrated as storing operating system 444, application programs 445, other program modules 446, and program data 447. Note that these components can either be the same as or different from operating system 434, application programs 435, other program modules 436, and program data 437. Operating system 444, application programs 445, other program modules 446, and program data 447 are given different numbers here to illustrate that they are at least different copies.

[0066] A user may enter commands and information into the computer 410 through input devices such as a keyboard 462 and pointing device 461, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 420 through a user input interface 460 that is coupled to the system bus 421, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB).

[0067] A monitor 491 or other type of display device is also connected to the system bus 421 via an interface, such as a video interface 490. In addition to the monitor, computers may also include other peripheral output devices such as speakers 497 and printer 496, which may be connected through an output peripheral interface 495.

[0068] The computer 410 operates in a networked environment using logical connections to one or more remote computers, such as a remote computer 480. The remote computer 480 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and as a function of its particular implementation, may include many or all of the elements described above relative to the computer 410, although only a memory storage device 481 has been illustrated in Fig. 4. The logical connections depicted in Fig. 4 include a local area network (LAN) 471 and a wide area network (WAN) 473, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

[0069] When used in a LAN networking environment, the computer 410 is connected to the LAN 471 through a network interface or adapter 470. When used in a WAN networking environment, the computer 410 typically includes a modem 472 or other means for establishing communications over the WAN 473, such as the Internet. The modem 472, which may be internal or external, may be connected to the system bus 421 via the user input interface 460, or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 410, or portions thereof, may be stored in the remote memory storage device. By way of example and not limitation, Fig. 4 illustrates remote application programs 485 as residing on memory device 481. The network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

## **Conclusion**

[0070] Although the systems and methods for verifying relevance between terms and Web site contents have been described in language specific to structural features and/or methodological operations or actions, it is understood that the implementations defined in the appended claims are not necessarily limited to the specific features or actions described. Accordingly, the specific features and actions are disclosed as exemplary forms of implementing the claimed subject matter.